

Alice

Learning to program: Part Two

Writing Your Own Methods

by Ruthie Tucker and Jenna Hayes

Under the direction of Professor Susan Rodger

Duke University, July 2008

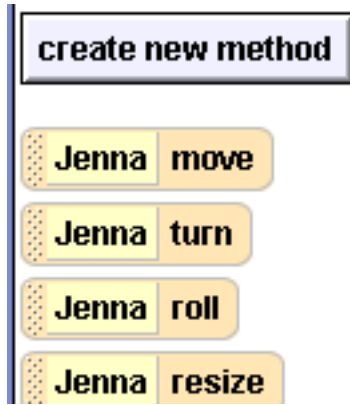


www.cs.duke.edu/csed/alice/aliceInSchools

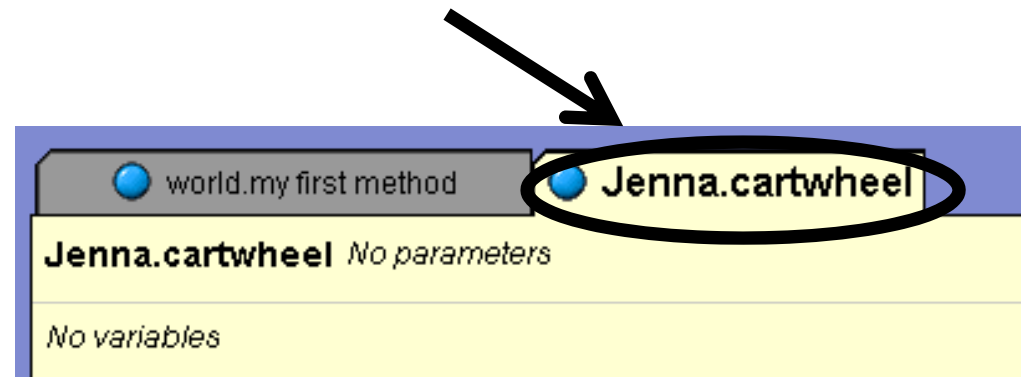
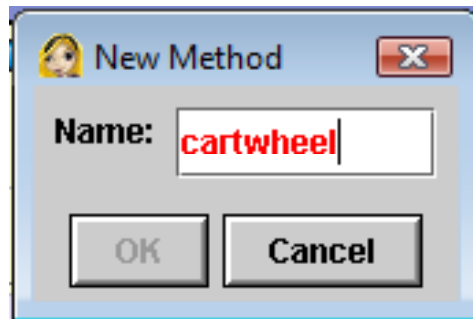
Making Your Own Methods

- What if we want our character to do something that he doesn't already know how to do?
- Now we will write a new method for Jenna to allow her to cartwheel.

Creating Your First Method

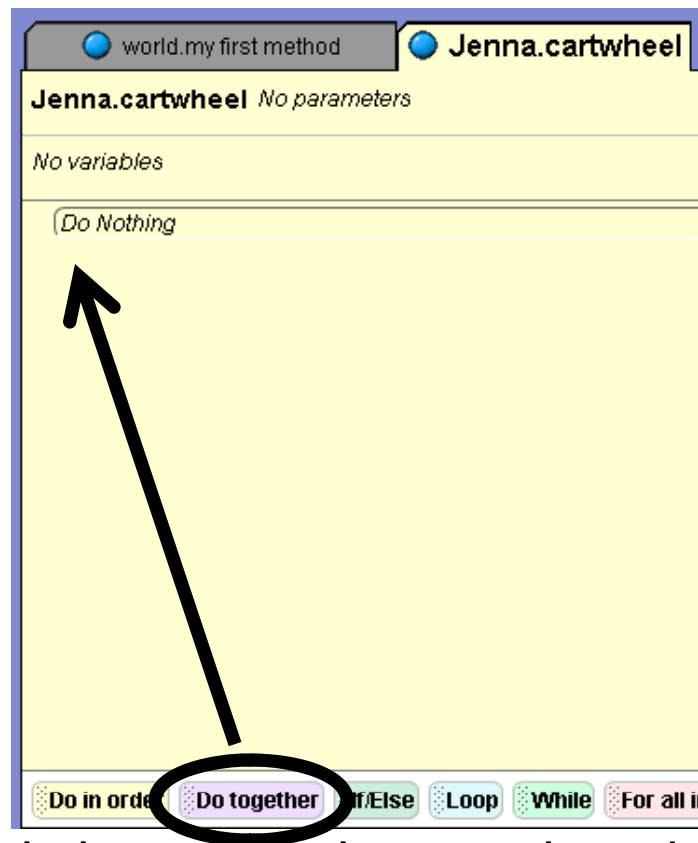


- Click on Jenna's name in the list on the left and scroll down to the area labeled "create new method".
- A gray box will pop up. Label it "cartwheel"
- Now, in the main box there should be a tab named "Jenna.cartwheel"



Creating Your First Method

- We need to code each of the character's body parts individually. Drag up a **Do together** from the bottom of the Alice window and drop it into your cartwheel method.



- Click on the body part that you want to code and look down at the series of commands
- Now look at the code on the next slide and make your code match it.

-Now click on the plus sign beside your character in the object list. The parts should expand out.

Drag and Drop in this code.

The screenshot shows a 3D environment with a horse and two people. On the left, a hierarchy of objects is listed: Jenna (LowerBody, UpperBody, Chest, RightUpperArm, LeftUpperArm, Neck), horse1, and Matt. The 'RightUpperArm's details' panel is open, showing methods like move, turn, roll, and resize. The 'roll' method is circled. The main workspace shows a 'Do together' block with two actions: 'Jenna.UpperBody.Chest.RightUpperArm roll left 0.18 revolutions' and 'Jenna.UpperBody.Chest.LeftUpperArm roll right 0.18 revolutions'. Arrows indicate the drag-and-drop process from the 'roll' method to the code block.

For example, click on **RightUpperArm** to see the methods for just this arm. Then, drag over the **roll** method, and set it to roll **left .18 revolutions**. Repeat this process for **LeftUpperArm**, choosing the opposite direction.

Final code

- Your final code should look like this. You should drag and drop in the rest of it.

Jenna.cartwheel *No parameters*

No variables

☐ Do together

Jenna.UpperBody.Chest.RightUpperArm ▾ roll left ▾ 0.18 revolutions ▾ more... ▾

Jenna.UpperBody.Chest.LeftUpperArm ▾ roll right ▾ 0.18 revolutions ▾ more... ▾

☐ Do together

Jenna ▾ roll left ▾ 1 revolution ▾ more... ▾

☐ Do together

Jenna.UpperBody.Chest.RightUpperArm ▾ roll right ▾ 0.18 revolutions ▾ more... ▾

Jenna.UpperBody.Chest.LeftUpperArm ▾ roll left ▾ 0.18 revolutions ▾ more... ▾

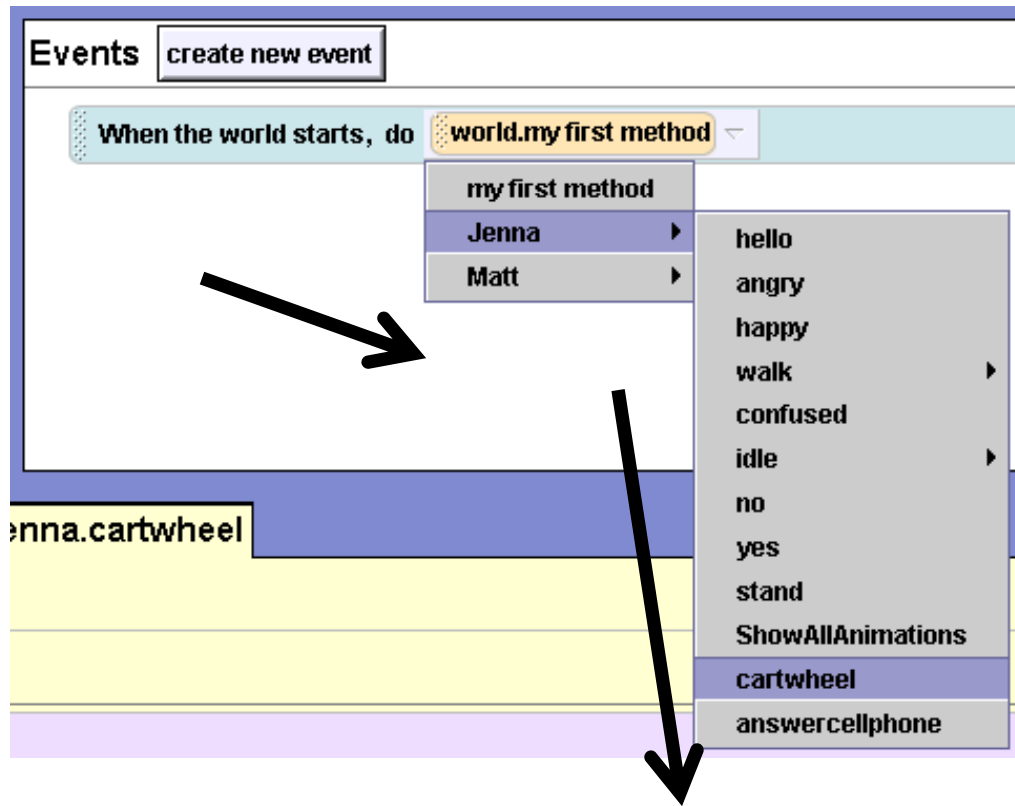
Now click play.



You should notice that nothing happens. Why is this? Look in the top right hand corner of your screen, and find where it says **when the world starts do**.

Notice that it says, do **world.my first method**. The method that you just wrote is called Jenna.cartwheel.

Playing Your New Method



Look at the right corner of your screen. There should be a section called “Events”.

Click on “When the World Starts”

Select “Jenna” and then “cartwheel”.

Now select Play

Your Character should animate into a cartwheel



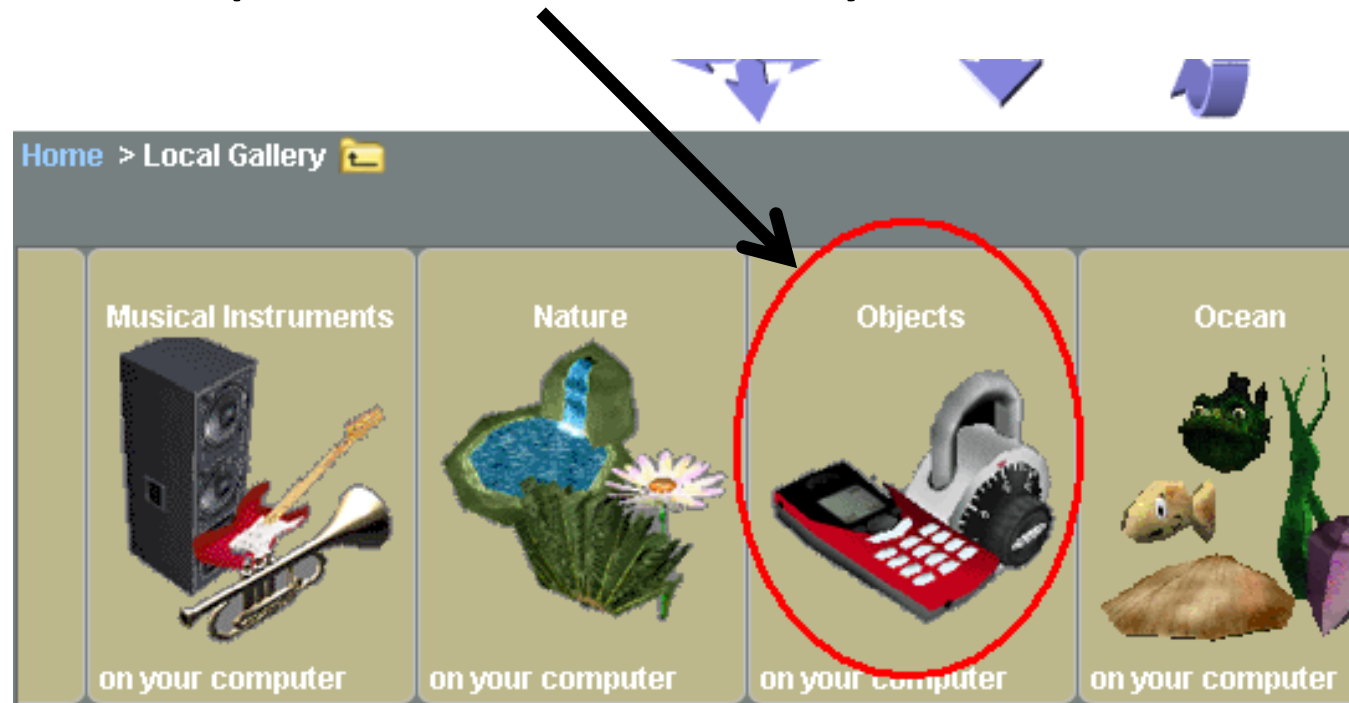
This is a good way to test a method that you have just written.

Methods (Continued)

- Now lets create a slightly more complicated method.
- Click on the “Create New Method” button again and name this one “answercellphone”.

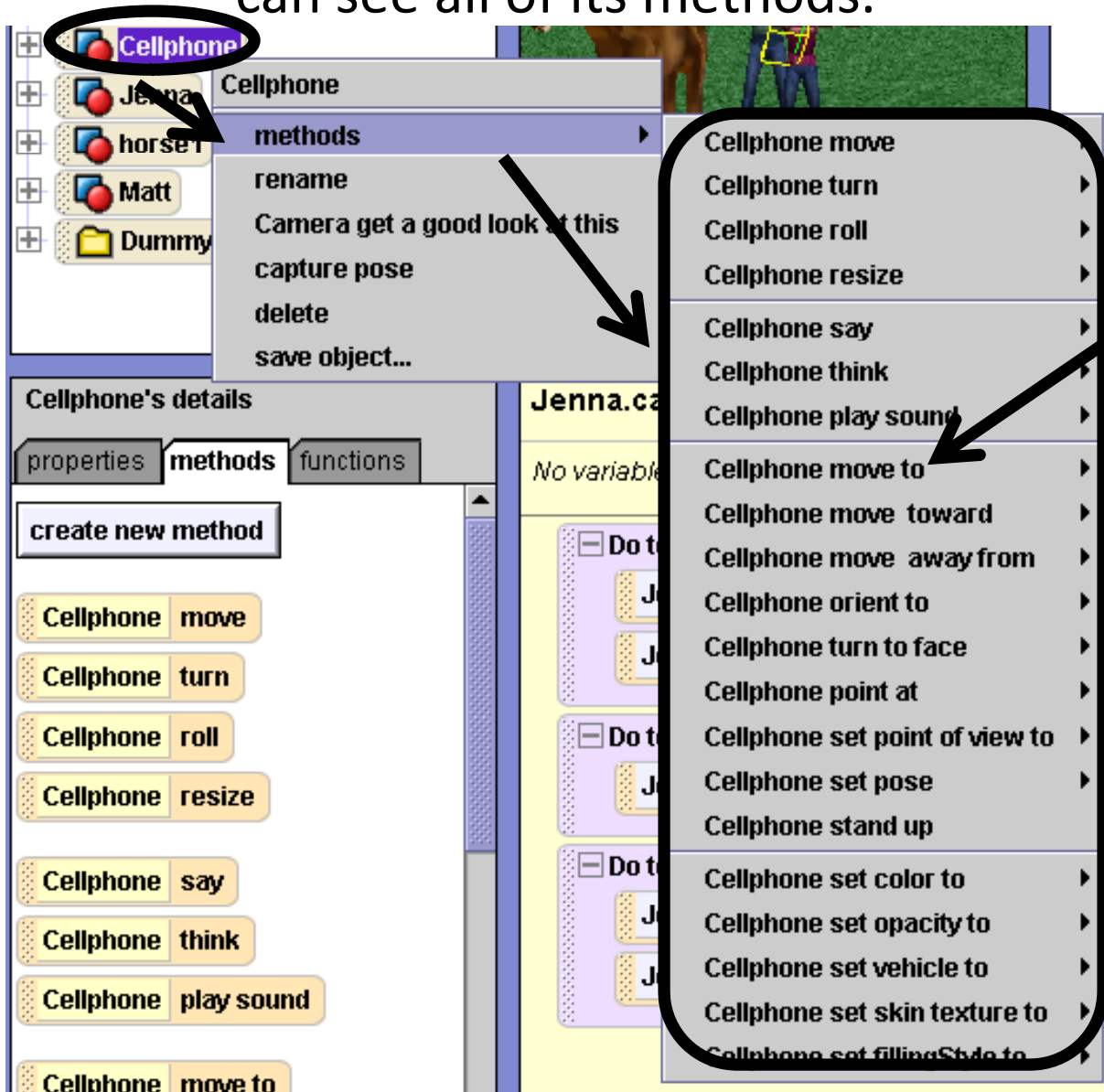
Methods (Continued)

- Click on “Add Objects” and scroll down to the “objects” folder. There should be an object labeled “cell phone”. Add it to your world.



Positioning the Phone

- If you right click on the cell phone in the object tree you can see all of its methods.



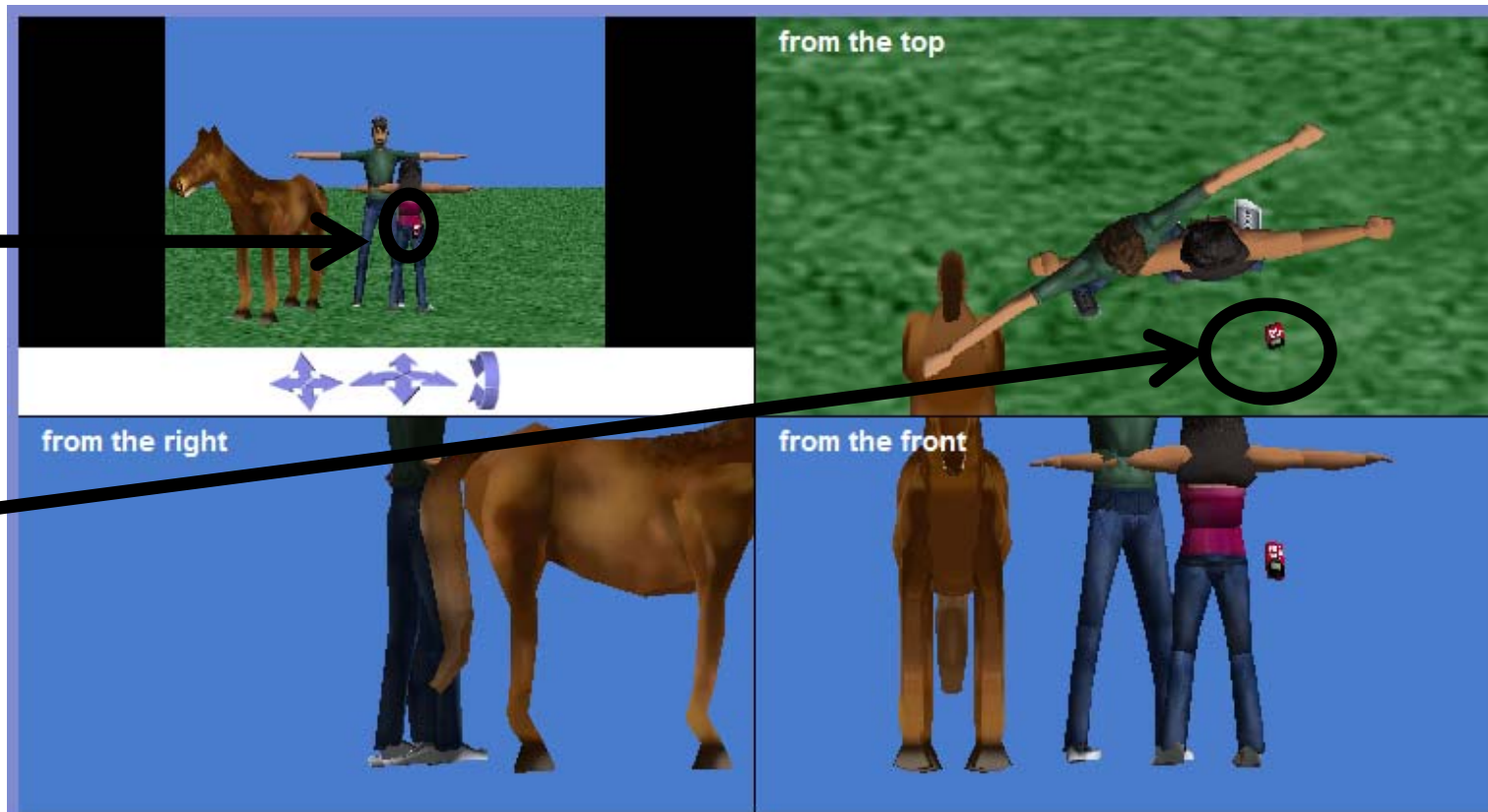
-Select **move to**, and have the phone move to Jenna.

-This will simply position your object in your viewing screen. It will not change your code that you have written.

Positioning the Object

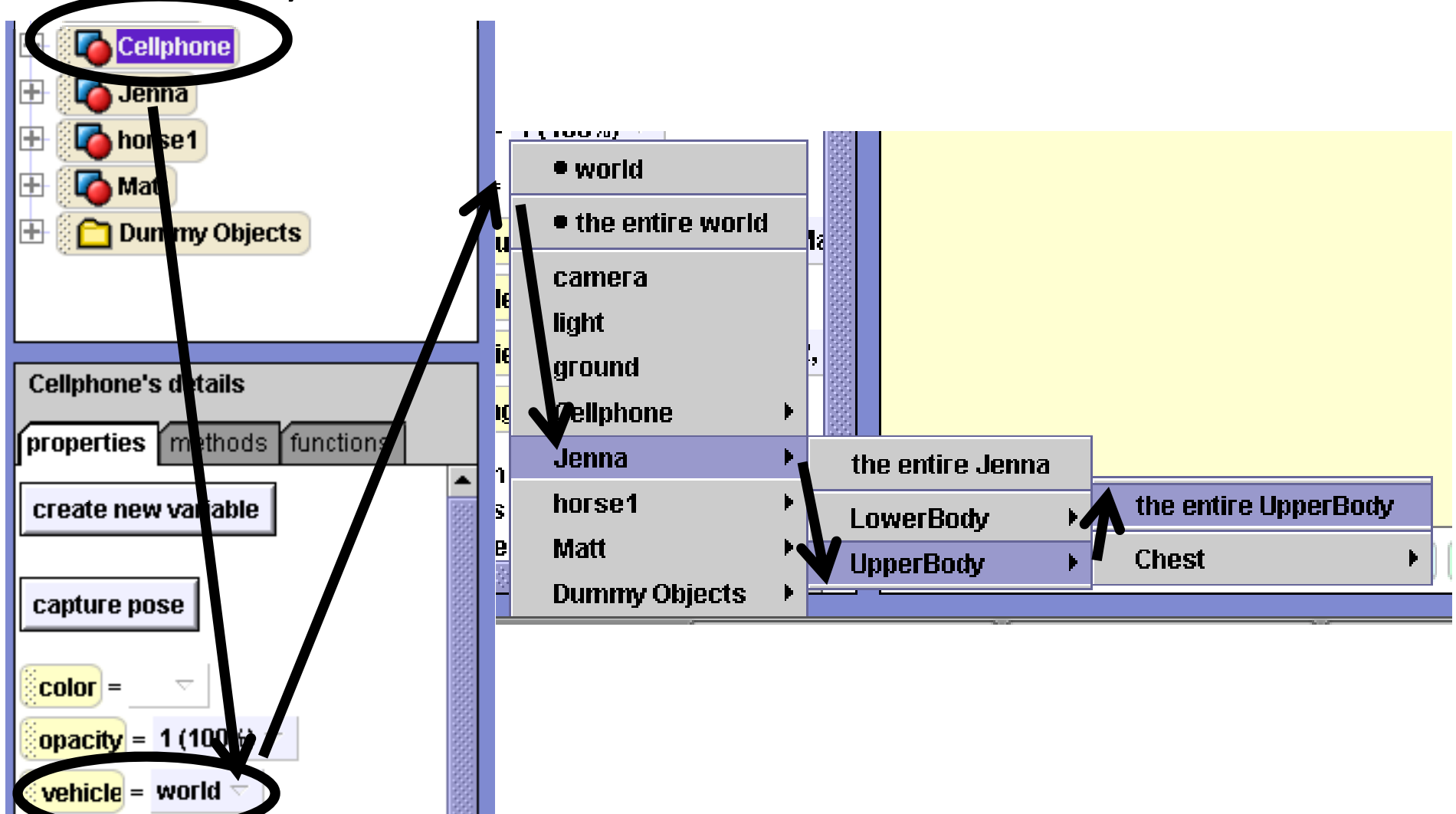
- Now use the object moving buttons to place the phone near the back pocket of your character.
 - You can turn your character around to do this.
- You should also try using quad view to make sure that the cellphone is very close to her pants, because it is hard to tell from the normal view.

Looks normal from here, but its actually far away.



Glue Cellphone to Pocket

- Now select the properties tab of “cell phone” and scroll down to the command called “vehicle”. Click on the down arrow next to the Vehicle button and set the Cell phone as a vehicle to the character’s upper body.

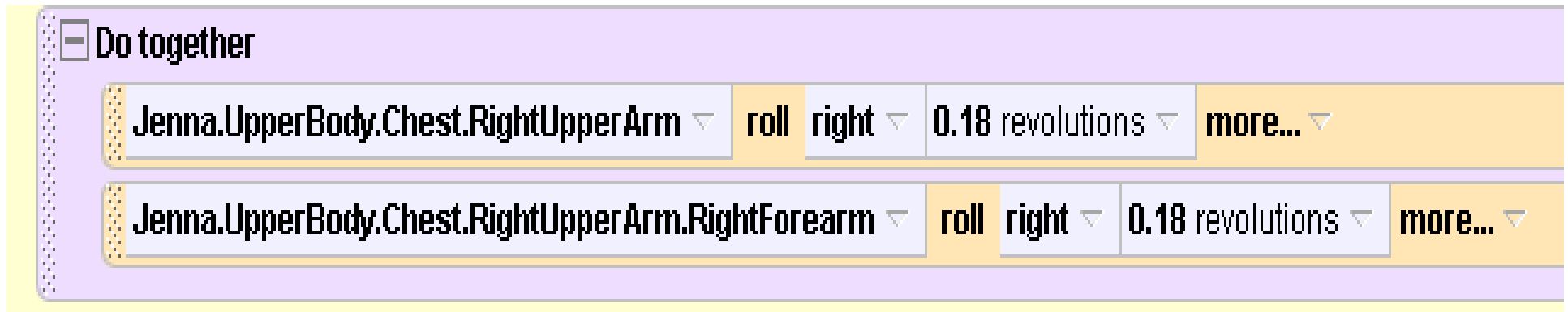


Vehicle Property

- The vehicle property basically allows you to glue one object to another object.
- If you set the cell phone as a vehicle to the character, you can move your character around and the cell phone will follow. Turn your character around with your object moving buttons to see the cellphone move with him or her.
- Now we can code the character to take the cell phone out of her back pocket.

Method “answercellphone”

- The first part is simply to code your characters arm to reach down and grab her cell phone



A Scratch code block titled "Do together" containing two movement blocks. The first block is for "Jenna.UpperBody.Chest.RightUpperArm" with the action "roll right" by "0.18 revolutions". The second block is for "Jenna.UpperBody.Chest.RightUpperArm.RightForearm" with the action "roll right" by "0.18 revolutions". Both blocks have a "more..." dropdown menu.

```
Do together  
  Jenna.UpperBody.Chest.RightUpperArm roll right 0.18 revolutions more...  
  Jenna.UpperBody.Chest.RightUpperArm.RightForearm roll right 0.18 revolutions more...
```

- Now go back to your vehicle property and drag it into the method to set the cell phone as a vehicle to your character's hand. The cell phone is now glued to your character's hand.



A Scratch code block with the action "set vehicle to" and the target "Jenna.UpperBody.Chest.RightUpperArm.RightForearm.RightHand.RightFingers". It has a "more..." dropdown menu.

```
Cellphone set vehicle to Jenna.UpperBody.Chest.RightUpperArm.RightForearm.RightHand.RightFingers more...
```

Method “answercellphone”

- Now code the arm to come up to your characters ear and have your character say “hello”.

Do together

Jenna.UpperBody.Chest.RightUpperArm ▾ roll left ▾ 0.18 revolutions ▾ more... ▾

Jenna.UpperBody.Chest.RightUpperArm.RightForearm ▾ roll left ▾ 0.5 revolutions ▾ more... ▾

Do together

Jenna ▾ say hello ▾ duration = 4 seconds ▾ more... ▾

Method “answercellphone”

Here is the completed method.

Jenna.answercellphone *No parameters*

No variables

[-] Do together

Jenna.UpperBody.Chest.RightUpperArm ▾ roll right ▾ 0.18 revolutions ▾ more... ▾

Jenna.UpperBody.Chest.RightUpperArm.RightForearm ▾ roll right ▾ 0.18 revolutions ▾ more... ▾

Cellphone ▾ set vehicle to Jenna.UpperBody.Chest.RightUpperArm.RightForearm.RightHand.RightFingers ▾ more... ▾

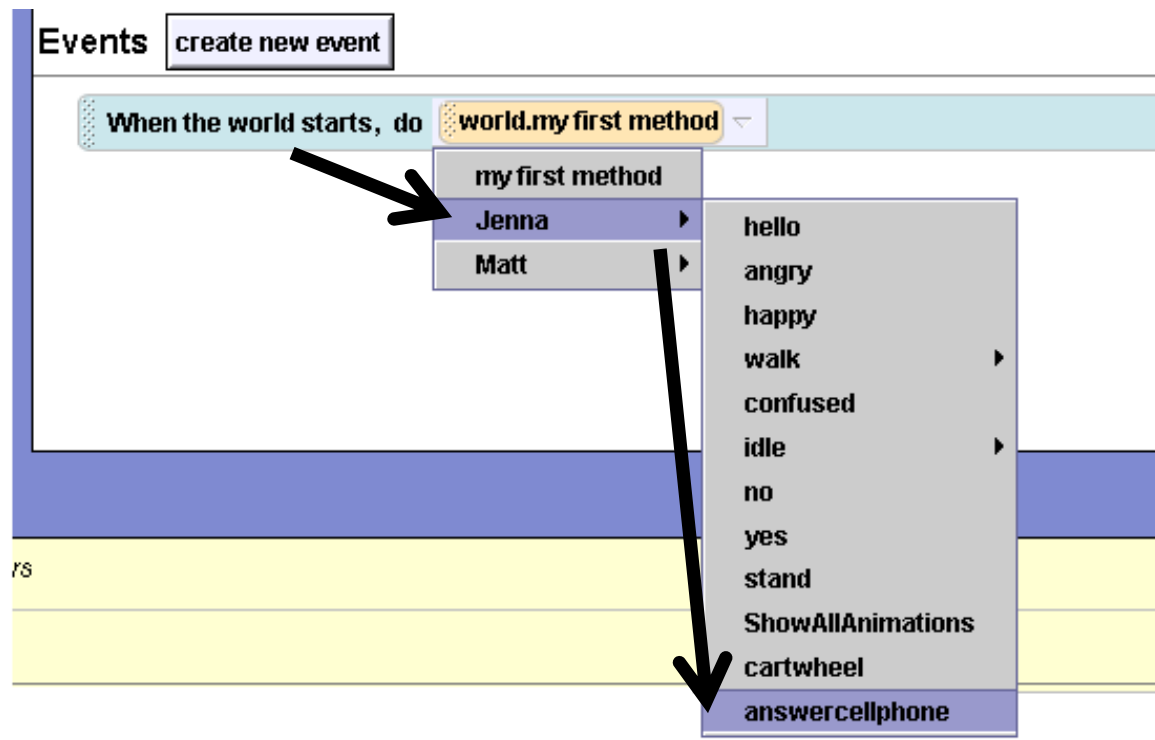
[-] Do together

Jenna.UpperBody.Chest.RightUpperArm ▾ roll left ▾ 0.18 revolutions ▾ more... ▾

Jenna.UpperBody.Chest.RightUpperArm.RightForearm ▾ roll left ▾ 0.5 revolutions ▾ more... ▾

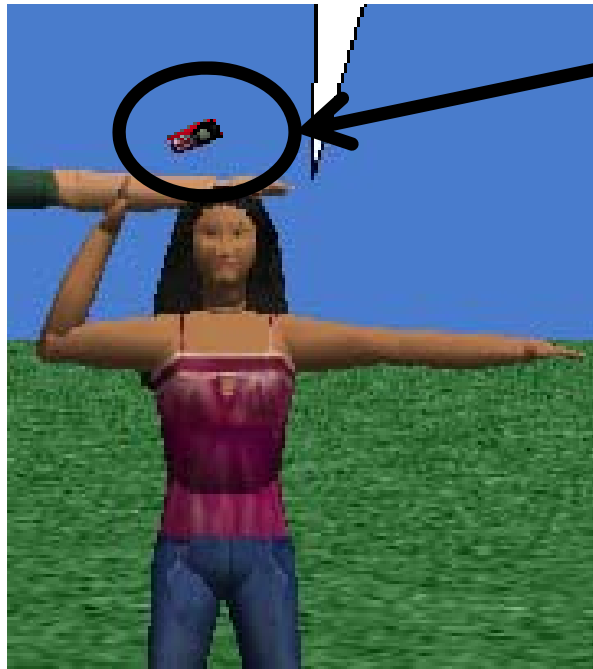
Jenna ▾ say hello ▾ more... ▾

Now you can try out your answercellphone method by going back to **when the world starts** and changing it to **Jenna.answercellphone**.



Press play to see what it looks like.

-Your cellphone method may still look a little bit wrong. If Jenna's phone looks like it is floating in the air, you should go back into quad view to reposition it so that it is right on her pocket. It may look right from the front, but when you view it from the side, you may see that the phone is actually several meters away from Jenna.



-If your phone is not close enough to Jenna's hand when she moves it up, for example if it is on her forearm, try using quad view to move the phone left and right to get the perfect position.

Method “Ride Horse”

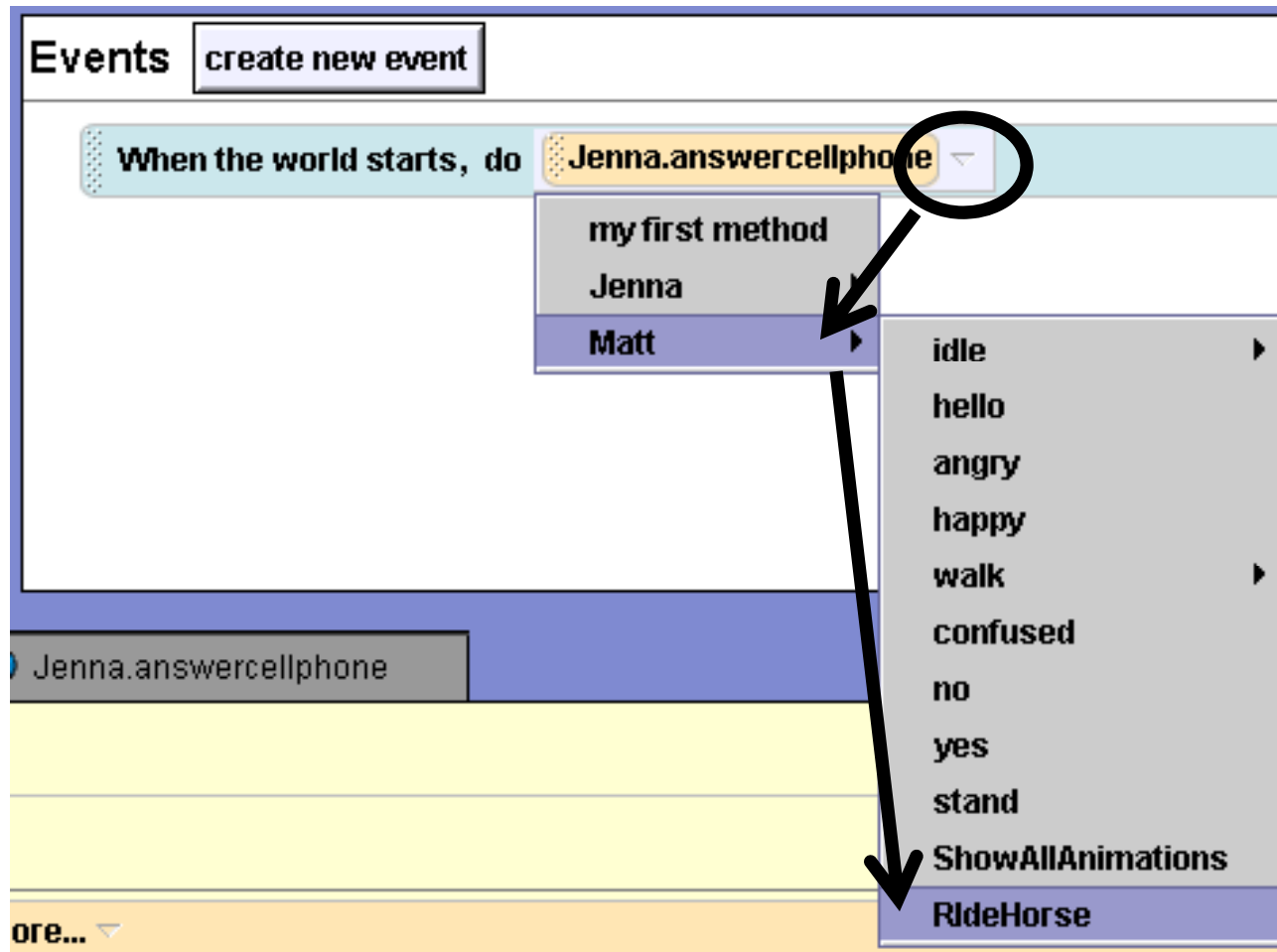
- Now lets create a method for the other character and the horse.
- Click on your other character in the object tree (we are doing it for Matt), and then click create new method. label the method “Ride Horse”.
- Lets break this down by steps
- First, we need the character to turn and face the horse



- Then we need the character to get on the horse
- Finally we need him to face forward again

Playing Your “Ride Horse” Method

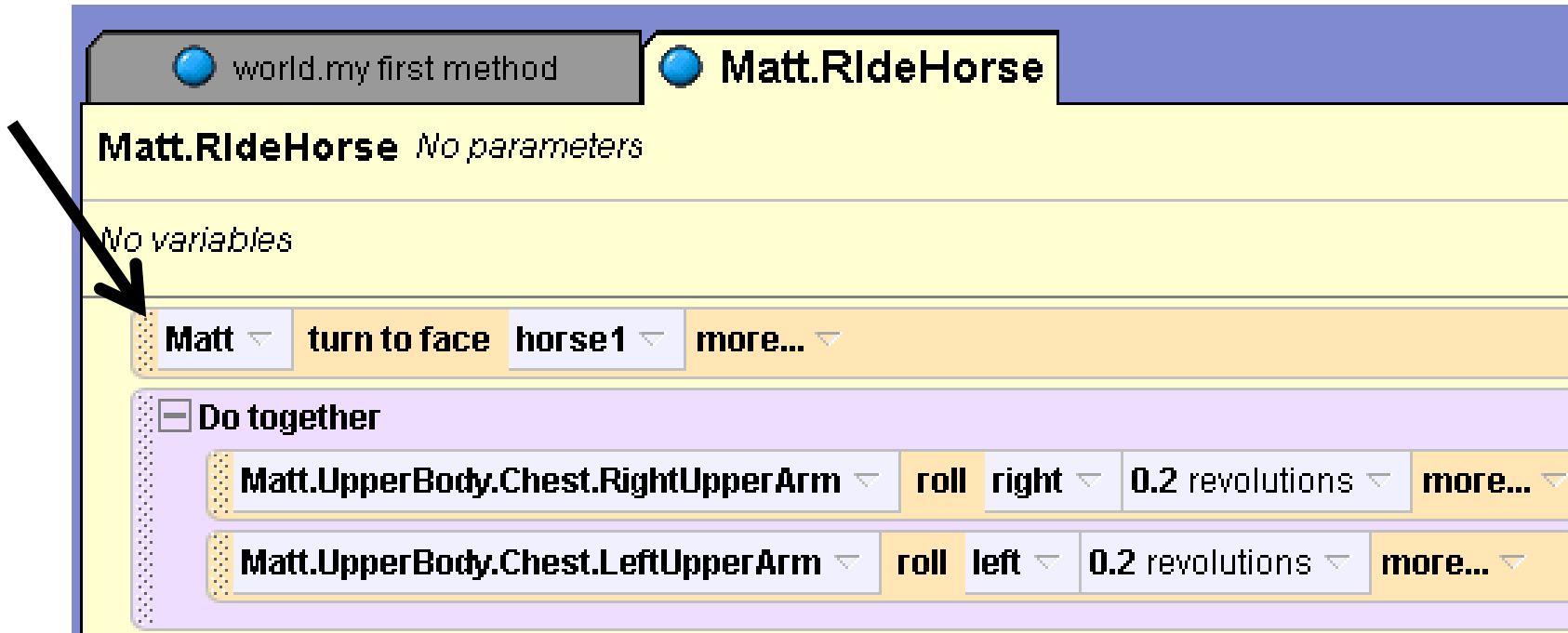
-In order to see your Ride Horse method when you play your world, you need to change your **when the world starts** event again. Change it from **answercellphone** to **Ride Horse**.



-Each time you add code to **Ride Horse** you should play your world to test it out.

Method “Ride Horse”

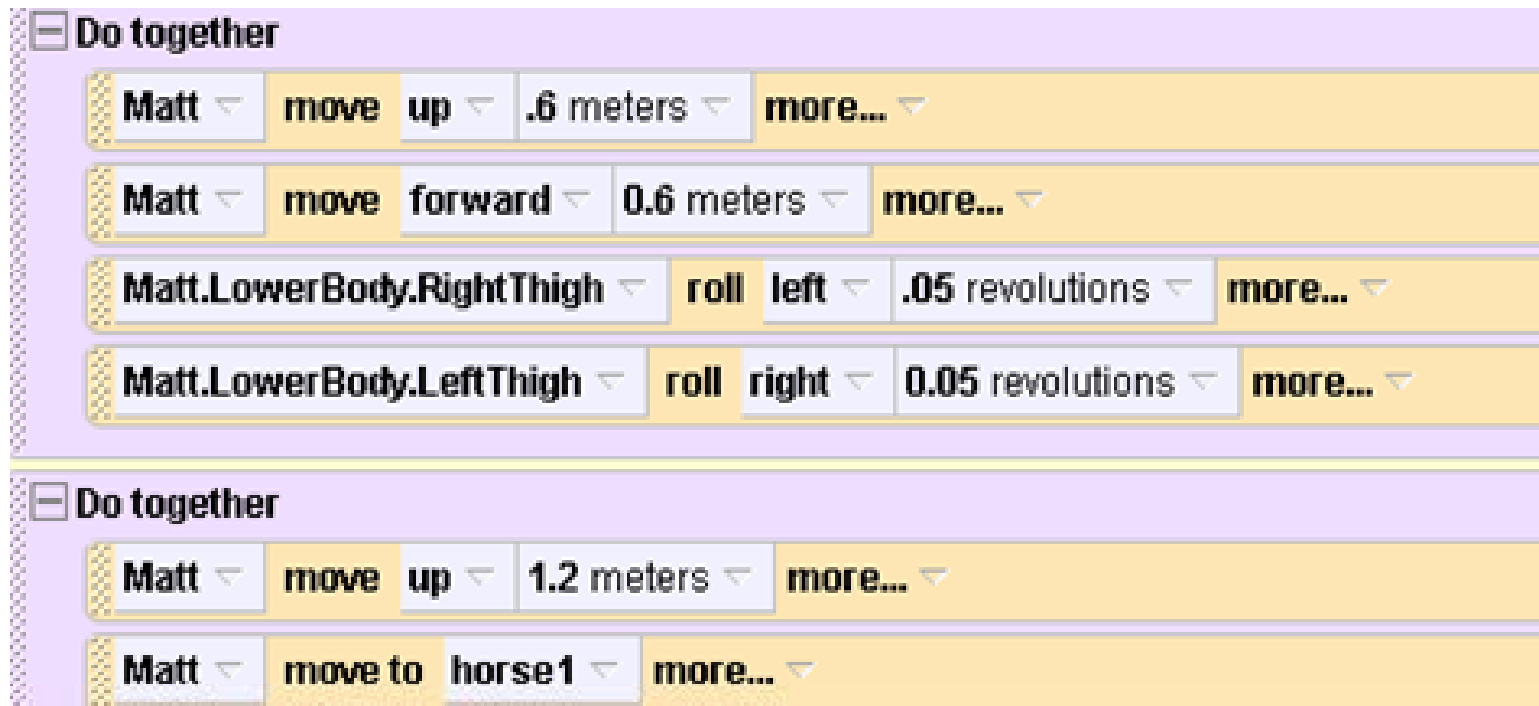
- For the first part we will simply write code to turn the character towards the horse and put his arms down.



Test this code by playing your world. If your characters arms do something other than moving down, you may have to try using a **turn** method instead of a **roll** method. This is because of the different sizes and shapes of the people-builder objects.

Method “Ride Horse”

- To code the character walking towards the horse, you simply need to move his limbs and keep playing it until you are happy with his movements. Add this code to the end of **Ride Horse**.



You may need to adjust these numbers according to the position of your character. If Matt does not go up high enough, make him move farther up. If he goes up too far, don't make him move quite as much.



If Matt looks like this, you may need to tell him to move up less, or move down more.



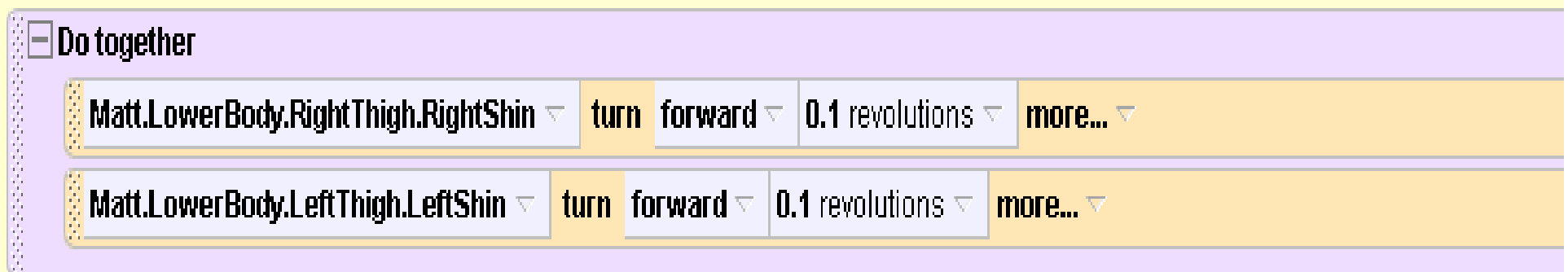
If Matt looks like this, you may need to tell him to move up more.

Method “Ride Horse”

- The final part will be to code our character to face the same direction as the horse.



- We can also tilt his shins back to look more realistic.



Add this code to the end of **Ride Horse**.

Finishing “Ride Horse”

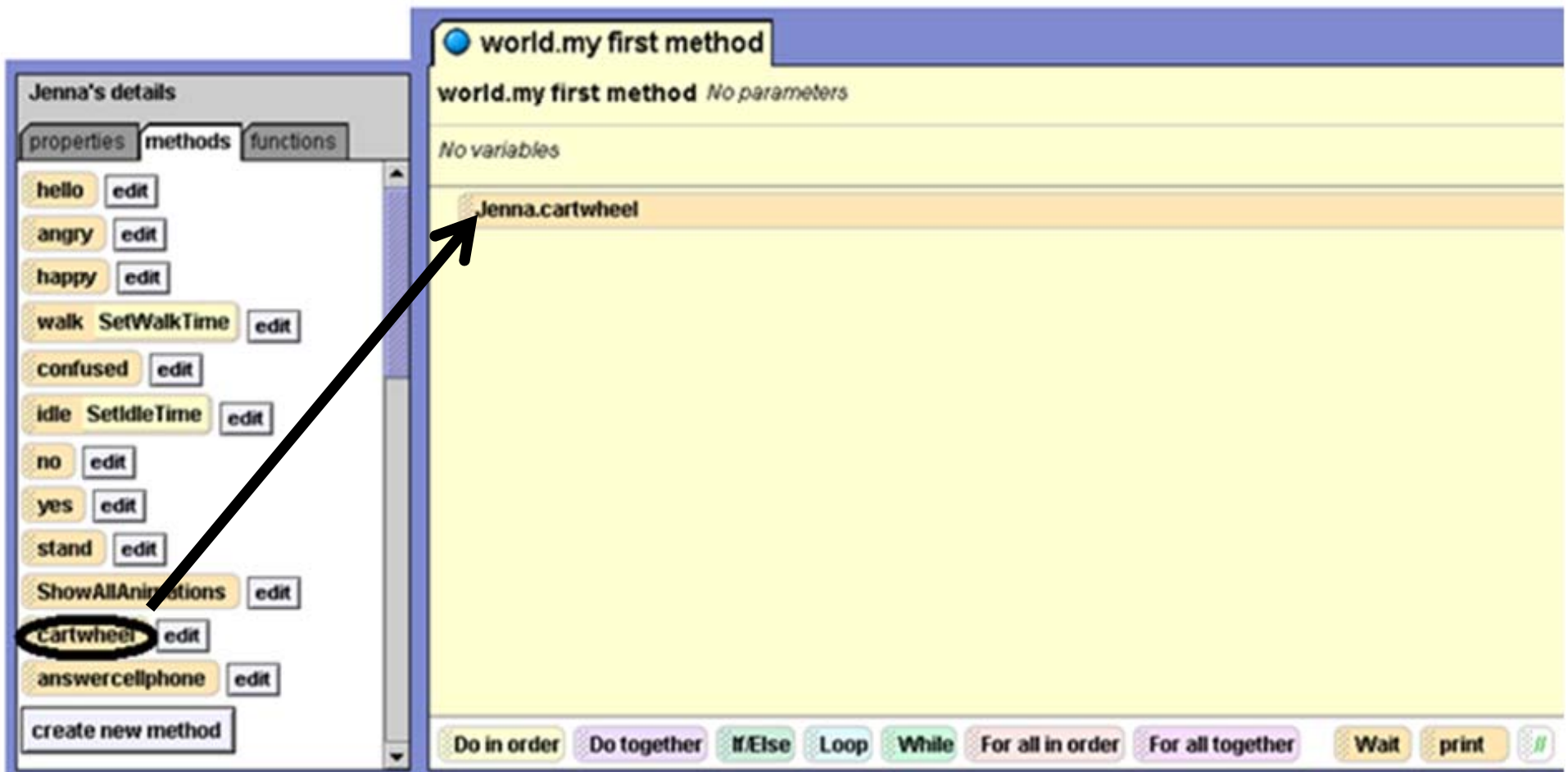


- Your final product should look something like this. Remember, your numbers may need to be different than the ones suggested here.
- Remember to keep playing your world as you code.
- If you code something wrong it is easy to see when you play the animation
- The “Undo” button is your friend!

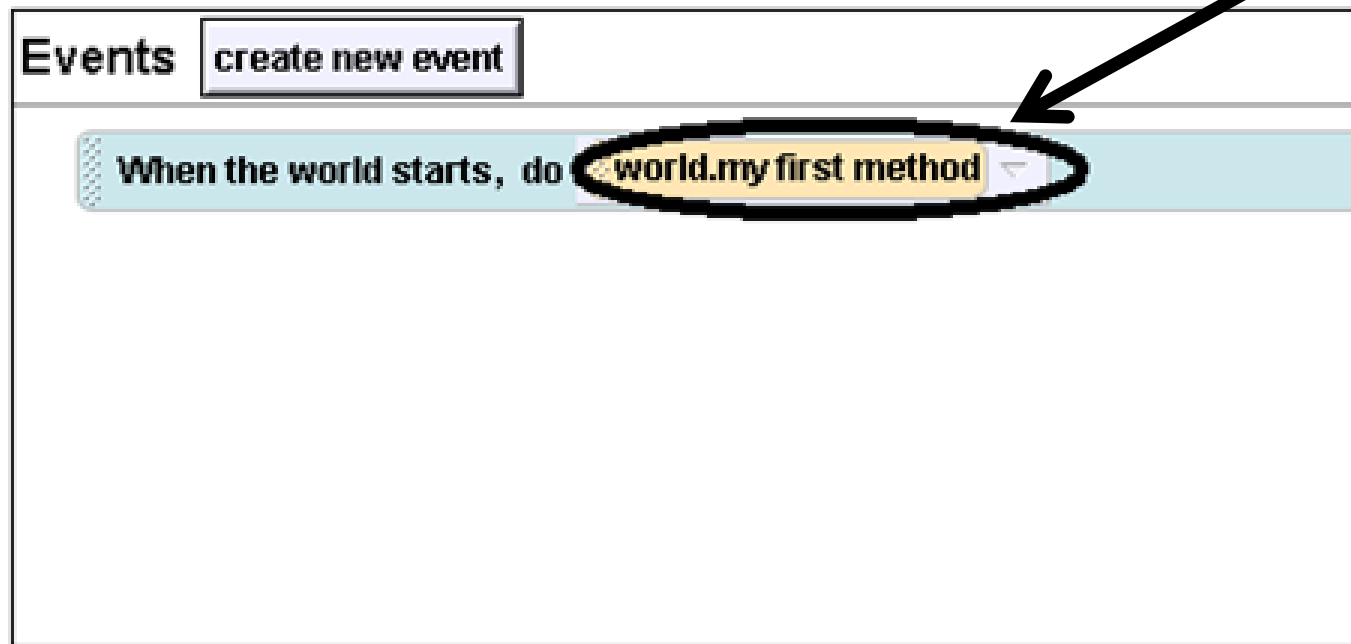
Now that you know how to write a method for a specific object, we are going to put these methods together to make a story. Click on the **world.my first method** tab on your screen. You may have code in there already from when you were testing out the he-builder/she-builder methods. You can keep it if you like, or erase it and add it back in later. These slides will start with a fresh, empty **my first method**.



This is where we will keep the parts of our story. First, we will use the cartwheel method that we taught to Jenna. Click on **Jenna** in the list of objects on the left side of the screen, and then the **methods** tab to find cartwheel again. Drag cartwheel into **my first method**.

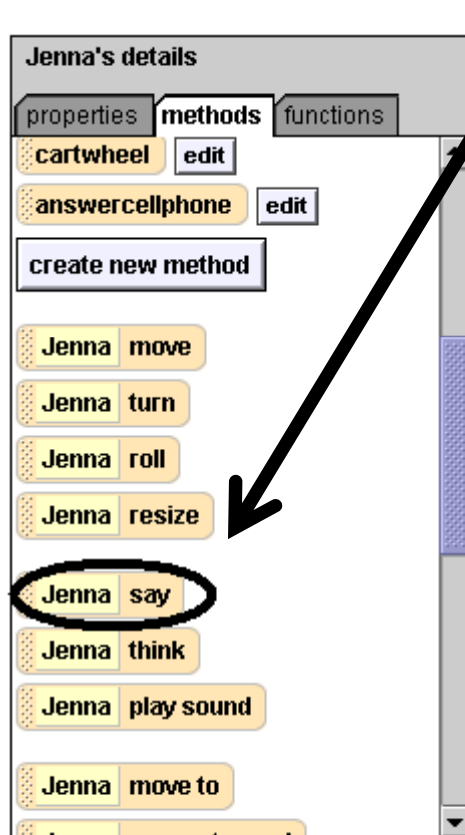


In the right hand corner of your screen find where it says **when the world starts do**, and change it to **world.my first method**. This way whenever you play your world it will play the story we will have in **my first method**.



Now play your world to see what it looks like so far. It should just show one of your characters cartwheeling.

Now we want to make our character say something. Look at the methods tab and scroll down until you see **say**.

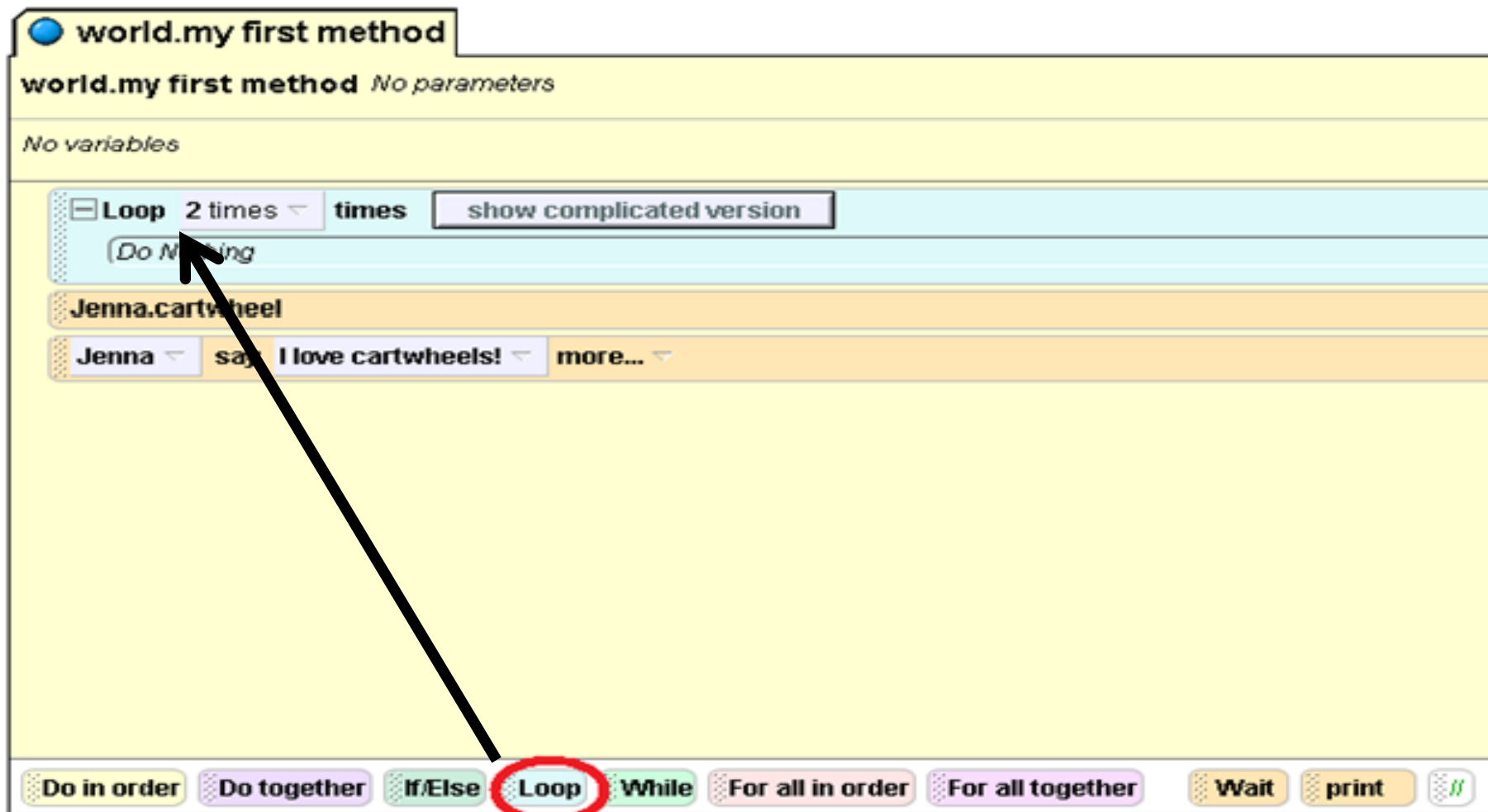


Now drag it into **my first method** under cartwheel. Click on **other...**, and you will have to type in something for your character to say. Type in something like "I love cartwheels!"



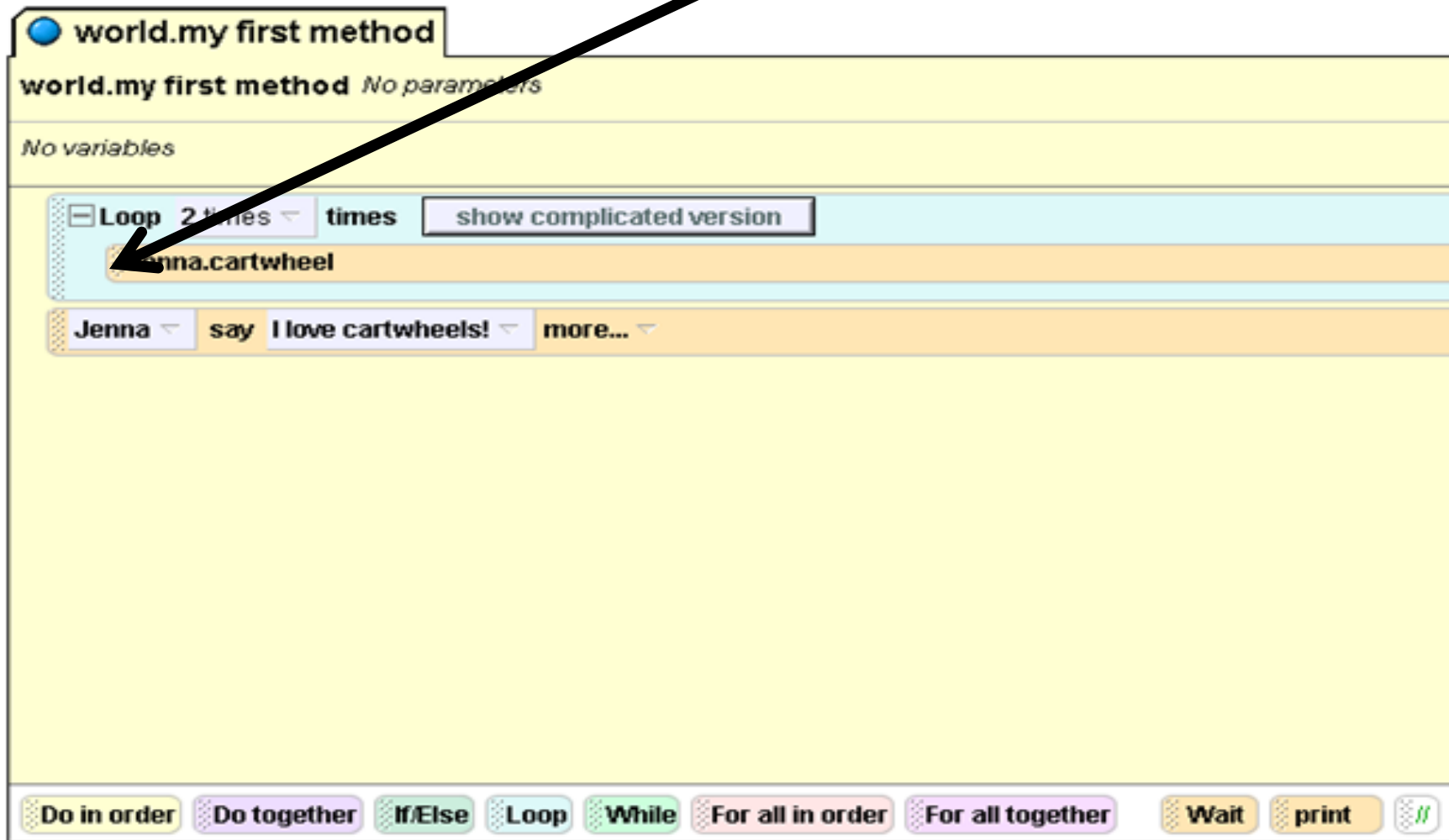
Play your world again to see the results.

Now we want to change it so that your character cartwheels twice in a row, to show just how much she loves cartwheels. To do this, we will use a **loop**. The loop button is located at the bottom of your method editor. Drag it into **my first method** above cartwheel. When you drop it select **2 times**.

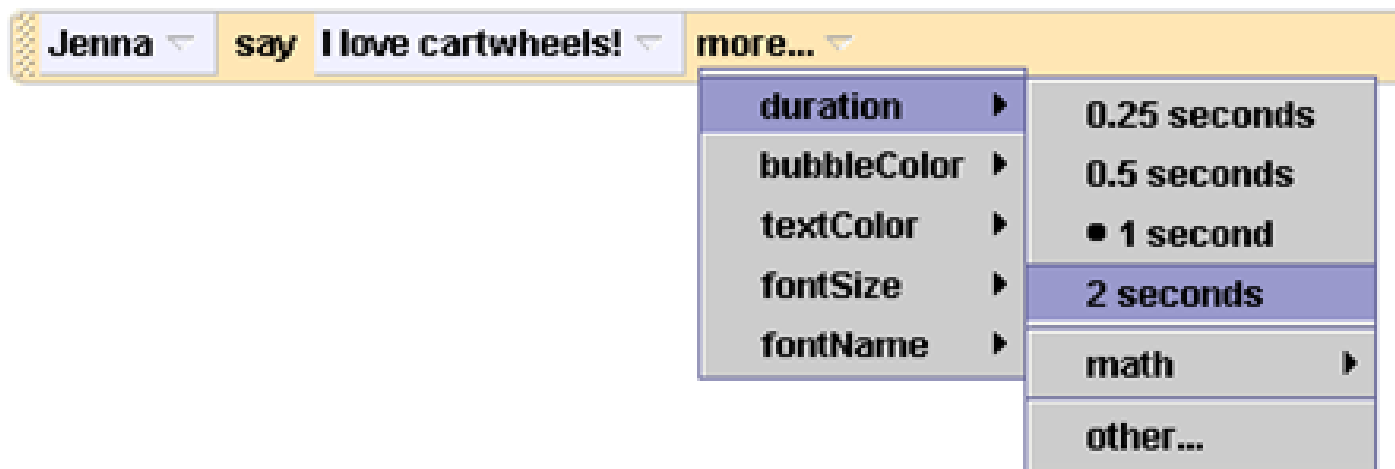


Now drag and drop your cartwheel command inside the **Loop**. In order to grab the cartwheel command to drag it, you must click on the left-most part of the method. Your cartwheel will now run 2 times! Play your world to see what happens.

Grab your method here to drag it.

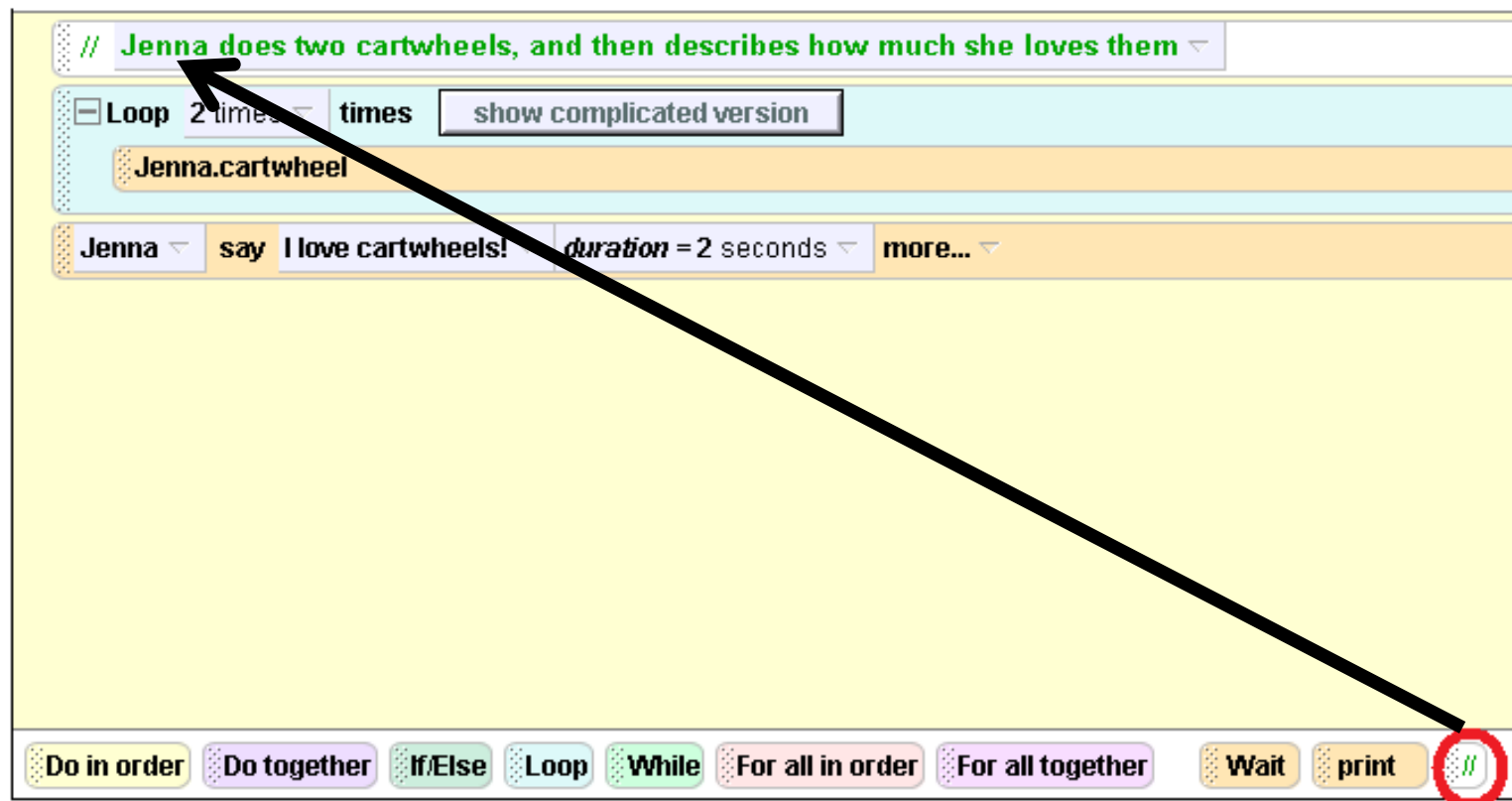


You may feel that your character's speech bubble disappears too quickly. If this is true, there is a way to make it last longer. Click on **more...** at the end of the say command. Then click on **duration**. As you can see, it lasts 1 second. You can make this longer if you wish, maybe 2 or 3 seconds.

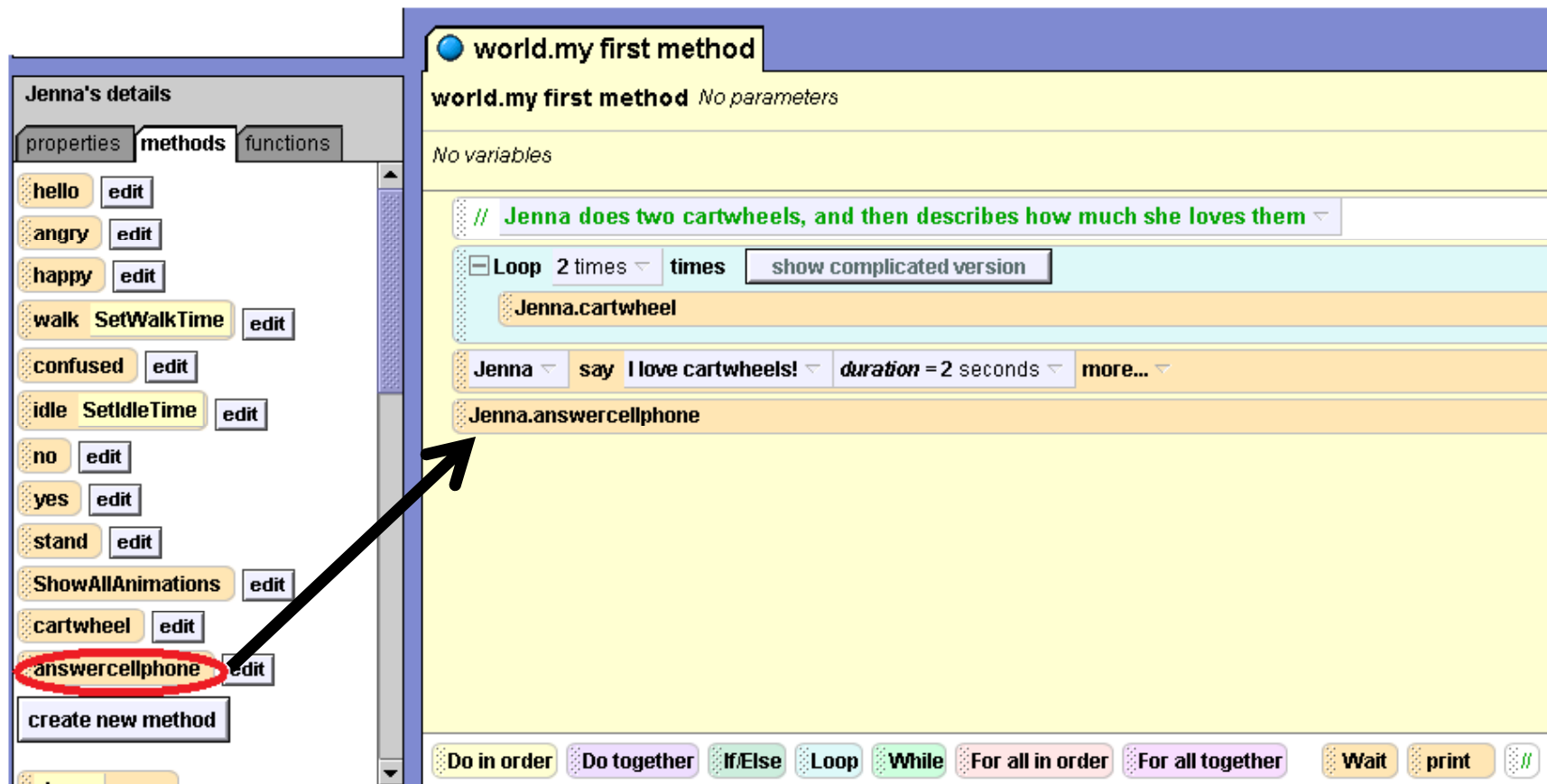


Play your world to test out different durations until you find one you like.

Now we are going to add **comments**. These are descriptions that you put in your code so that you, or anyone else that reads your code, can see what it does. The comment button is located at the bottom of the method editor. Drag it into the top of **my first method** and type a quick description of what your code does.



Now we will make your character answer her phone. Find the **answercellphone** method and drag it into **my first method**. Play your world again to see your changes.



Now we'll add one more method to the story. Click on Matt, your other character, in the list of objects, and look at his methods until you find the RideHorse method. Add it to **my first method**. Then add another comment above **answercellphone** that describes the rest of your code.

The screenshot shows a programming environment with a sidebar on the left and a main script area on the right.

Sidebar (Matt's details):

- Properties: idle, SetIdleTime, edit
- Methods: hello, edit; angry, edit; happy, edit; walk, SetWalkTime, edit; confused, edit; no, edit; yes, edit; stand, edit; ShowAllAnimations, edit; **RideHorse, edit** (circled); create new method
- Objects: Matt, move

Main Script Area (world.my first method):

- Script title: world.my first method
- Script content:
 - // Jenna does two cartwheels, and then describes how much she loves them
 - Loop 2 times times show complicated version
 - Jenna.cartwheel
 - Jenna say I love cartwheels! duration = 2 seconds more...
 - // Jenna answers her phone and says hello, and then Matt gets on the horse.
 - Jenna.answercellphone
 - Matt.RideHorse** (circled)

Arrows indicate the relationship between the 'RideHorse' method in the sidebar and the 'RideHorse' block in the script area. One arrow points from the sidebar to the script area, and another points from the script area back to the sidebar. A third arrow points from the 'RideHorse' block in the script area to the 'RideHorse' method in the sidebar.

Congratulations on creating a great story! Now we are ready to move on to Part 3 to learn new things about changing the camera view.

